

## 論文の内容の要旨

論文題目	Debugging and Testing Concurrent Programs with Efficient Test Case Generation 効率良いテストケース生成による並行処理プログラムのデバッグとテスト
学位 申請者	セティアディ    テオドラス    エリック Setiadi            Theodorus        Eric

マルチスレッドからなる並行処理プログラムのデバッグは逐次処理プログラムに比較してはるかに難しい。その理由の一つは、エラーの再現が難しいからである。トレースを取るために、並行処理プログラムを再実行したり、コードに何らかの措置を施すと、実行のタイミングが変化したり、異なった実行パスを取るようになる。即ち、元のエラーが発生した厳密なタイミングは未知である。エラーを再現するためには、たとえ入力変数の値が同一でも、インターリーブを変化させた多くのテストケースの実行が必要となる。しかも、それらを全て実行できるとは限らない。

本論文では、マルチスレッド並行処理プログラムにおいて、異なったスケジュールや割込タイミングの違いに起因するエラー、特に、レース条件の再現のための方法を提案する。実行トレースから得られた限定された情報の範囲で、全ての可能なテストケースを生成し、レース条件を検出するデバッグ／テストシステムを提案した。これまで、partial order reductionを用いたテストケース削減に関する多くの研究が存在するが、レース条件の検出という意味ではまだ冗長な部分が存在する。本論文での目的は、効率的にレース条件を検出することにある。そのために、3つの手法を提案する。

一つ目は、レース条件検出能力を維持しながら、冗長なテストケースを削除することである。提案手法の独創性は、分岐に与えるインターリーブの影響に注目し、トレース情報から得られたデータフロー情報を活用し、分岐結果に影響を与えるインターリーブのみを特定することにある。既存手法の多

くは、共有変数に影響を与える全てのインターリーブを特定することを行っている。この違いにより、提案手法はテストに必要なインターリーブの数をさらに大きく削減することができる。同一のロック状態と共有変数へのアクセス状態を有する実行パスを“レース同等グループ”として1つのグループにまとめ、そのグループを構成するメンバーの中の1個のみをテストすることにより、レース条件テストに要する労力を軽減する。さらに、提案手法は既存の実行トレースのモデルを拡張し、lock/unlockとwait/notify 依存性により実行不可能なインターリーブを特定できる。こういった実行不可能なインターリーブをテストケースから省くことにより、テストケース削減に大きく寄与できることを示した。

二つ目は、テストケースを生成するために必要なメモリ容量を削減することである。既存のreachability testing手法は、実行されることがないインターリーブに対してもテストケースを生成する。本提案手法はデータ依存性を解析することにより、ロック状態と共有変数へのアクセス状態に影響を与えるインターリーブのみ生成する。これにより、テストケースを生成するためのグラフの大きさを削減でき、その維持のためのメモリ容量を削減できる。実際、Apache Commons Poolに対する実験結果では、提案手法によってグラフのサイズが990ノードから4ノードに減少した。

三つ目は、繰り返しテストの労力を軽減することである。本提案手法を含むデバッグ／テスト手法においては、プログラムの実行トレースを繰り返し取ることによりプログラム誤りの有無と、その誤りの検出を行う。本提案手法では以前の実行トレースとテストの結果を利用して、レース条件をチェックする手間を軽減する。従来法では、テスト毎に全体の実行トレースを取り、レース条件をチェックする必要がある。本提案手法は、生成されたインターリーブに基づき、ロック状態を変更する可能性がある実行トレースの部分を識別することができる。そのことにより、ロック状態を変更する可能性のある実行トレースの部分のみを再チェックの対象とすることにより、チェックの数を削減できた。

以上の新たな改善策の導入により、本提案手法は与えられた入力値に対して、可能なインターリーブの網羅的なテストを、はるかに少ない労力でもって達成する。インターリーブに起因するプログラムエラーの有無、エラーが発生するインターリーブ（パス）、ロックが正しく行われていない共有変数へのアクセスに関する情報がプログラマに提供される。

## 論文審査の結果の要旨

学位申請者氏名      SETIADI THEODORUS ERIC

審査委員主査      多田    好克

委員      本多    弘樹

委員      大森    匡

委員      大須賀    昭彦

委員      田原    康之

委員      ※前川    守

マルチスレッドからなる並行処理プログラムのデバッグとテストは逐次処理プログラムに比較してはるかに難しい。その理由の一つは、エラーの再現が難しいからである。本論文では、マルチスレッド並行処理プログラムにおいて、異なったスケジュールや割込タイミングの違いに起因するエラー、特に、レース条件の再現のための方法を提案する。実行トレースから得られた限定された情報の範囲で、全ての可能なテストケースを生成し、レース条件を検出するデバッグ／テストシステムを提案した。本論文での目的はレース条件検出能力を維持しながら、そうした冗長なテストケースを削除することにある。

第1章ではマルチスレッドからなる並行処理プログラムのデバッグとその問題点を示す。マルチスレッド並行プログラムの挙動はタイミングに依存して時折エラーとなるといった非決定性があるため、デバッグは困難である。トレースを取るために、並行処理プログラムを再実行したり、コードに何らかの措置を施したりすると、実行のタイミングが変化したり、異なった実行パスを取ったりすることとなる。エラーを再現するためには、たとえ入力の値が同一でも、インターリーブを変化させた多くのテストケースの実行が必要となる。しかも、それらを全て実行できるとは限らない。

第 2 章では並行プログラムのデバッグングの手段として決定的再実行とテストケース削減を用いた関連既存システムのサーベイを示す。これまで、partial order reductionを用いたテストケース削減に関する多くの研究が存在するが、レース条件の検出という意味ではまだ冗長な部分が存在する。

第 3 章では基本的な用語と定義を示す。マルチスレッドの並行実行における主要な異常の 1 つであるレース状態と、レース状態を解決する手段としての並行制御機構について述べる。テストケース数を削減するために、「レース同等」と呼ぶ新たな概念を導入する。2 つの実行は、共有変数へのアクセスとそのロック操作が同じであるとき「レース同等」であると言う。分岐が共有変数へのアクセスまたはロック順序に影響する場合、これがレース同等グループの違いの原因になり得る。

第 4 章では本論文の問題設定と、解決手法の要件を示す。デバッグングにおいては、静的手法はしばしば実際の実行では起こり得ないエラーも検出するので、動的手法が必要である。動的手法を実用的にするためには、必要なテストケース数とレース状態検出の手間を最小にしなければならない。

第 5 章ではテストケースとして必要な、互いに異なるインターリーブの数を削減する新たな手法を提案する。この手法は、実行トレースから得られるデータフローに基づいて、実行分岐に影響を与えるインターリーブだけを特定する。これにより、テストに必要なインターリーブの数が大幅に削減できる。データフローは、“use-define”連鎖の定義をマルチスレッド並行プログラム用に拡張することにより計算する。同一のロック状態と共有変数へのアクセス状態を有する実行パスを“レース同等グループ”として 1 つのグループにまとめ、そのグループを構成するメンバーの中の 1 個のみをテストすることにより、レース条件テストに要する労力を軽減する。さらには、レース状態とデッドロックを検出するためのオーバーヘッドを削減する手法を示す。この削減は、過去に観測されたすべてのロック構造のログを記録する事により可能となる。もし、新たに観測されたロック構造がログに記録されたどれかと同じであれば、レース検出を行なう必要がなくなる。

第 6 章では Java 言語における実装手法について論じ、いくつかの実験結果を、既存の Thread-Pair-Interleaving (TPAIR) 手法と比較しながら示す。オープンソースプログラムの Apache Commons Pool の実験結果においては、テストケースの数が 23 から 2 に減少される。

第 7 章では提案手法の効率について論じる。テストケース数の削減量は、対象プログラムの構造に依存する。提案手法では、一般に、分岐やループのための条件文がインターリーブの違いによってあまり影響を受けない場合に、削減処理が効率的に行える。

第 8 章では結論を述べる。提案手法の独創性は分岐に与えるインターリーブの影響に注目し、分岐結果に影響を与えるインターリーブのみを特定し、トレース情報から得られたデータフロー情報を活用することにある。既存手法の多くは、共有変数に影響を与える全てのインターリーブを特定することを行っている。この違いにより、提案手法はテストに必要なインターリーブの数をさらに大きく削減することができる。実験結果により、提案手法は冗長なインターリーブを特定することができ、テストケース削減に大きく寄与できることを示した。

以上のような内容に鑑み、本論文は博士（工学）の学位申請論文として十分な価値を有すると判断した。